# Stochastic Link-Fault-Tolerant Routing in Hypercubes

Lam Boi Ngoc, Bui Thi Thuan, Yuki Hirai, and Keiichi Kaneko

Abstract—Recently, parallel processing systems are enthusiastically studied, and many topologies for their interconnection networks have been proposed. A hypercube was one of such popular topologies for interconnection networks, and it still provides fundamental structures for practical supercomputers such as the NASA Pleiades and the NOAA Zeus and theoretical hierarchical topologies based on it. In this paper, we propose a link-fault-tolerant routing algorithm in hypercubes based on routing probabilities. The probability represents routing ability of a node for an arbitrary node located at a specific distance. Each node selects one of its neighbor nodes to send the message by considering their routing probabilities. We also conducted a computer experiment to evaluate performance of our algorithm.

*Index Terms*—Routing probability, fault tolerance, dependable computing, interconnection network, parallel processing.

# I. INTRODUCTION

Recently, parallel processing systems are enthusiastically studied since sequential processing is reaching its performance limitation. There are many topologies proposed for interconnection networks of parallel processing systems. A hypercube was one of the most popular topologies for interconnection networks of parallel processing systems [1]. Its symmetric and recursive structure and the low diameter compared to its degree have attracted much attention. As a result, several commercial and research parallel systems have adopted this topology [2]-[6]. Although, latest parallel computers do not adopt a hypercube for their topologies, a hypercube provides fundamental structures for practical supercomputers such as the NASA Pleiades and the NOAA Zeus [7] and theoretical hierarchical topologies based on it [8]-[12].

In a parallel processing system, many nodes collaborate together by communication among nodes based on message passing. Therefore, efficiency of message passing has huge impact on performance of parallel processing. The role of a routing algorithm is to specify the path to send a message from a source node to a destination node. Hence, efficient message routing is one of the most important issues in parallel processing systems.

Because the sizes of applications addressed by parallel

Lam Boi Ngoc is with the Faculty of Mathematics and Computer Science, Ho Chi Minh University of Science, Ho Chi Minh City, Viet Nam (e-mail: boingoclam5893@gmail.com). processing systems are increasing, higher performance systems connecting more nodes are needed. However, according to increase of the number of nodes in the system, the probability of existence of faulty elements also increases. Therefore, it is necessary to design routing algorithms with assumption that there exist faulty elements in a parallel processing system. For a source node and a destination node, a fault-tolerant routing algorithm attempts to find a fault-free path between them whose length becomes as short as possible. Although there are several approaches to address this problem, an efficient routing algorithm must satisfy the following conditions.

First, for a pair of the source and destination nodes, the algorithm must find a fault-free path between them. Second, if a node has knowledge about faulty elements in the network, it can select the optimal path. Because of resource limitation, that is, time and space complexities, each node cannot identify all faulty elements. Instead, each node stores small amount of information, which is called limited global information. When the system is started or rebooted, at each node, the limited global information is collected by exchanging status information with its neighbor nodes and it is stored in the node.

Fault-tolerant routing in the state-of-the-art hypercubebased hierarchical topologies makes use of fault-tolerant routing in a hypercube [13]. Therefore, in this paper, we assume that an n-dimensional hypercube  $Q_n$  has faulty elements, and propose a link-fault-tolerant routing algorithm that finds a fault-free path for an arbitrary pair of nodes. In the algorithm, time and space complexities of information stored in each node must be the polynomial orders of n. In general, a link-fault-tolerant routing algorithm is more generic than a node-fault-tolerant routing algorithm since a faulty node can be treated by regarding all of the links incident to a faulty node as faulty.

The rest of this paper is structured as follows. First, in Section II, we survey the related works. Next, we define the requisite terminology and notations for discussion in Section III. Then, in Section IV, we introduce the routing probability, its approximate value, and a simple calculation method for approximate values. In addition, in Section V, we propose a link-fault-tolerant routing algorithm based on routing probabilities. In Section VI, we evaluate performance of the algorithm by a computer experiment. Finally, we give conclusion and a future work in Section VII.

# II. RELATED WORKS

In these two decades, research on node-fault-tolerant routing in hypercubes attracts much attention, and many research activities for this problem have been reported. Chiu and Wu proposed an efficient node-fault-tolerant routing

Manuscript received September 30, 2014; revised March 18, 2016. This work was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant No. 25330079.

Bui Thi Thuan, Yuki Hirai, and Keiichi Kaneko are with Graduate School of Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan (e-mail: k1kaneko@cc.tuat.ac.jp).

algorithm by recursively classifying each non-faulty node into a safe, ordinary unsafe, or strongly unsafe node based on the classification of its neighbor nodes [14]. To improve this algorithm, Chiu and Chen introduced routing capabilities, which are obtained by classifying safe nodes according to the Hamming distances to destination nodes [15]. Wu also proposed a similar node-fault-tolerant routing algorithm independently by introducing safety vectors [16]. In addition, Kaneko and Ito proposed a node-fault-tolerant routing algorithm based on classification of ordinary and strongly unsafe nodes with respect to Hamming distances and an efficient calculation method for classification [17].

All of the above approaches are based on information whether a message will be surely delivered to the destination node or not. On the other hand, Al-Sadi *et al.* and Duong and Kaneko proposed node-fault-tolerant routing algorithms based on the probability that a message will be delivered from the source node to the destination node with the path whose length is equal to the Hamming distance between the nodes [18]-[20]. Although these algorithms are based on distinct approaches, their results are very similar. In addition, Duong and Kaneko improved their algorithm by introducing directed probabilities, which at each node make use of the information of the neighbor node from which the message is sent to the node [21].

Unfortunately, all of the related works are intended for node-fault-tolerant routing in a hypercube. Hence, in this research, we enhance the node-fault-tolerant routing algorithm by Duong and Kaneko [20] so that it can tolerate link faults. Moreover, we carry out a computer experiment to evaluate its performance.

#### III. PRELIMINARIES

In this section, we define a hypercube and introduce necessary notations. For two nodes  $a = (a_1, a_2, ..., a_n)$  and  $b = (b_1, b_2, ..., b_n)$ , the Hamming distance between them H(a, b) is defined by the number of positions at which the corresponding bits  $a_i$  and  $b_i$   $(1 \le i \le n)$  are different.

## A. Hypercubes Definition

An *n*-dimensional hypercube  $Q_n$  is an undirected graph, and it consists of  $2^n$  nodes. Each node a is an *n*-bit sequence  $(a_1, a_2, ..., a_n)$  where  $a_i \in \{0, 1\}$   $(1 \le i \le n)$ , and  $a_i$  is called the bit of (i - 1)th dimension. For two nodes a and b in  $Q_n$ , there is a link between them (a, b) if and only if the Hamming distance between them H(a, b) is equal to 1. The neighbor node of aalong a dimension j  $(0 \le j \le n - 1)$  is  $a \oplus 2^j$ .

Fig. 1 shows a 4-dimensional hypercube  $Q_4$ .



Fig. 1. A 4-dimensional Hypercube  $Q_4$ .

In general, a path in a graph is represented by an alternating sequence of nodes and links  $a_1$ ,  $(a_1, a_2)$ ,  $a_2$ , ...,  $a_{k-1}$ ,  $(a_{k-1}, a_k)$ ,  $a_k$ . For simplicity, we use the notation of  $a_1 \rightarrow a_2 \rightarrow ... \rightarrow a_k$  in this paper. The length of a path, P, is the number of links included in the path, and it is denoted by L(P). If  $Q_n$  is fault-free, the length of a shortest path between two nodes a and b is equal to H(a, b).

#### B. Faulty Link Set Definition

For a node a in  $Q_n$ , the set of node N(a) defined by

$$N(a) = \{n \mid n \in Q_n, H(a, n) = 1\}$$

is called the set of neighbor nodes of *a*.

In a hypercube  $Q_n$  with a set of faulty links F, for a source node s and a destination node d, a fault-tolerant routing algorithm finds a fault-free path between s and d. To represent link faults, we introduce a function  $\gamma$  that is defined as follows for two adjacent nodes a and b:

$$\gamma(a,b) = \gamma(b,a) = \begin{cases} 0, ((a,b) \in F) \\ 1, ((a,b) \notin F) \end{cases}$$

We assume that each non-faulty node a in  $Q_n$  can identify  $\gamma(a, n)$  in constant time for an arbitrary neighbor node n.

# C. Neighbor Nodes

In  $Q_n$ , for two nodes a and b, a set of preferred neighbor nodes of a for b is denoted by  $N_0(a, b)$ , and it is defined by  $N_0(a, b) = \{n \mid n \in N(a), H(n, b) = H(a, b) - 1\}$ . In addition, a set of spare neighbor nodes of a for b is denoted by  $N_1(a, b)$ , and it is defined by  $N_1(a, b) = \{n \mid n \in N(a), H(n, b) = H(a, b) + 1\}$ . Note that  $N(a) = N_0(a, b) \uplus N_1(a, b)$  holds where the operator  $\bowtie$  represents the direct sum of two sets.

Fig. 2 illustrates the set of preferred neighbor nodes and the set of spare neighbor nodes. In  $Q_n$ , the number of nodes with Hamming distance h from a node a is equal to  ${}_nC_h$ . Moreover, for two nodes a and b in  $Q_n$ , if H(a, b) = h,  $|N_0(a,b)|=h$  and  $|N_1(a, b)| = n - h$  hold. For example, in a 4-dimensional hypercube  $Q_4$  shown in Fig. 1., the set of preferred neighbor nodes of a node 0010 for a node 1100 is  $N_0(0010, 1100) = \{0000, 0110, 1010\}$ , and the set of spare neighbor nodes of 0010 for 1100 is  $N_1(0010, 1100)=\{0011\}$ . The Hamming distance between these two nodes 0010 and 1100, H(0010, 1100) = 3, and it is equal to  $|N_0(0010, 1100)|$ .



#### IV. ROUTING PROBABILITIES

In this section, we give the concept of routing probabilities. In an *n*-dimensional hypercube  $Q_n$  with a set of faulty links *F*, let us consider that a node *a* has a message. Then,  $P_h(a)$ , the routing probability of a with respect to a Hamming distance h, represents the probability of existence of a fault-free path of length h between a and b for an arbitrary node b such that h=H(a, b).

Since it is difficult to calculate the routing probabilities exactly, we use the following approximate values.

#### A. Routing Probability with Hamming Distance

For a node a in an n-dimensional hypercube  $Q_n$  with a set of faulty links F, the approximate value of  $P_h(a)$ , a routing probability of a with respect to a Hamming distance h, is defined as follows:

$$P_{h}(a) = \begin{cases} 1 , h = 0 \\ \sum_{I \subset N(a), |I| = h} \max_{n \in I} \{\gamma(a, n), P_{h-1}(n)\} \\ n C_{h} \end{cases} (1)$$

Definition (1) for  $P_h(a)$  has the following meanings. First, an arbitrary node can deliver a message to itself with probability 1. Next, from N(a), the set of neighbor nodes of a node a, pick up a set of h nodes I, and calculate the maximum value of the approximate routing probabilities with respect to a Hamming distance (h - 1) of the elements of I. Their expected value is the approximate routing probability of the node a with respect to the Hamming distance h where we regard the approximate routing probabilities of the neighbor nodes of a connected to a by faulty links zero by introducing function  $\gamma$ .

In the rest of this paper, we call an approximate routing probability just a routing probability. To introduce a simplified calculation method, we prove the following theorem.

#### B. Routing Probability Calculation

In an *n*-dimensional hypercube  $Q_n$  with a set of faulty links *F*, for a node *a* and a Hamming distance *h*  $(1 \le h \le n)$ ,

$$P_{h}(a) = \left(\sum_{k=1}^{n} k_{-1} C_{h-1} p_{k}\right) / {}_{n} C_{h}$$
(2)

Holds where  $p_1 \le p_2 \le ... \le p_n$  are obtained by sorting  $\gamma(a, a \oplus 2^0) P_{h-1}(a \oplus 2^0)$ ,  $\gamma(a, a \oplus 2^1) P_{h-1}(a \oplus 2^1)$ , ...,  $\gamma(a, a \oplus 2^n) P_{h-1}(a \oplus 2^{n-1}) P_{h-1}(a \oplus 2^{n-1})$  in ascending order.

(**Proof**) In (1), there is a definition of  $P_h(a)$ ,  $p_k = \max_n \in I\{\gamma(a, n) P_{h-1}(n)\}$  holds if and only if  $p_k \in \bigcup_n \in I\{\gamma(a, n) P_{h-1}(n)\}$  where  $\bigcup_n \in I\{\gamma(a, n) P_{h-1}(n)\} \subset \{p_1, p_2, ..., p_k\}$  holds. Therefore, the number of occurrences that  $p_k$  becomes the maximum value is equal to  $_{k-I}C_{h-I}$ . Thus, the theorem holds.

From equation (2), we can obtain Procedure RP to calculate routing probabilities. Fig. 3 shows the procedure. Each node runs a program based on Procedure RP simultaneously, and calculates its routing probabilities.

For example, given a 4-dimensional hypercube  $Q_4$  with a set of 4 faulty links (0000, 1000), (0100, 0110), (0110, 1110), and (1001, 1011) shown in Fig. 4, Table I shows the product of the values of the function  $\gamma$  and the routing probabilities that each node collects from neighbor nodes. From Table I, because nodes 0100 and 1110 are connected to a node 0110 by faulty links, their routing probabilities are stored in the node 0110 as zero. Let us consider a case where a message is

sent from the node 0010 to the node 1100. Then, the Hamming distance between the node 0010 and the node 1100 is equal to 3, and the set of the preferred neighbor nodes is  $N_0(0010, 1100)=\{0000, 0110, 1010\}$ .

procedure RP(a)

begin  $P_0(a) := 1;$ send  $P_0(a)$  to each  $x \in N(a);$ for h := 1 to n do begin for each  $x \in N(a)$  do receive  $\gamma(a, x)P_{h-1}(x);$ sort  $\gamma(a, x)P_{h-1}(x)$   $(x \in N(a))$ to obtain  $p_1 \le p_2 \le ... \le p_n;$   $P_h(x) := \sum_{k=1}^n k_{-1}C_{h-1}p_k / {}_nC_h;$ send  $P_h(a)$  to each  $x \in N(a)$ end; for each  $x \in N(a)$  do

receive  $\gamma(a, x)P_n(x)$ ;

end

For the node in the set of the preferred neighbor nodes, the products of the routing probabilities with respect to the Hamming distance 2 and the values of the function  $\gamma$  are  $\gamma(0010, 0000)P_2(0000) = 0.96, \gamma(0010, 0110)P_2(0110) = 0.83, \gamma(0010, 1010)P_2(1010) = 0.88$ . Hence, it is appropriate to send the message to the node 0000, which has the maximum value. On the other hand, if the message is sent to the node 0110 without using the information, it would be impossible to send the message to the destination node 1100 with a shortest path because every shortest path to 1100 includes a faulty link.

The following theorem is useful in link-fault-tolerant routing based on routing probabilities.

#### C. Theorem of Routing Nodes Existence

In an *n*-dimensional hypercube  $Q_n$  with a set of faulty links *F*, for two arbitrary nodes *s* and *d* such that h = H(s, d), if there exists a fault-free path between *s* and *d*, there exists a node *a* such that  $P_{h-1}(a) > 0$  in  $N_0(s, d)$ , the set of the preferred neighbor nodes of *s* for *d*, or there exists a node *b* such that  $P_{h+1}(b) > 0$  in  $N_1(s, d)$ , the set of the spare neighbor nodes of *s* for *d*.

(**Proof**) Let  $c_0 \rightarrow c_1 \rightarrow \dots \rightarrow c_m$  be a fault-free path between sand d where  $c_0 = s$ ,  $c_m = d$ ,  $c_i \neq c_{i+1}$  ( $0 \le i \le m-1$ ),  $c_{i-1} \neq c_{i+1}$  $(1 \le i \le m-1)$ , and  $h \le m$ . If  $c_1 \in N_0(s, d)$ , consider a sub path  $c_0 \rightarrow c_1 \rightarrow \dots \rightarrow c_h$ . Then, from (1),  $P_0(c_h) > 0$  holds. Moreover, for an arbitrary r ( $0 \le r \le h-1$ ), if  $P_{h-r-1}(c_{r+1}) > 0$ ,

$$P_{h-r}(c_r) = \sum_{I \subset N(c_r), |I| = h-r} \max_{n \in I} \{\gamma(c_r, n) \cdot P_{h-r-1}(n)\} / {}_n C_{h-r}$$
  

$$\geq \gamma(c_r, c_{r+1}) \cdot P_{h-r-1}(c_{r+1}) / {}_n C_{h-r}$$
  

$$= P_{h-r-1}(c_{r+1}) / {}_n C_{h-r} > 0$$

holds. Hence,  $P_{h-1}(c_1) > 0$  holds. If  $c_1 \in N_1(s, d)$ , consider a sub path  $c_0 \rightarrow c_1 \rightarrow \dots \rightarrow c_{h+2}$ . Then, from similar discussion to the case of  $c_1 \in N_0(s, d)$ ,  $P_{h+1}(c_1) > 0$  holds.

From this theorem, for two arbitrary nodes *s* and *d* such that h = H(s, d), if there is not a node  $a \in N_0(s, d)$  such that

 $P_{h-1}(a) > 0$  nor a node  $b \in N_1(s, d)$  such that  $P_{h+1}(b) > 0$ , there is no fault-free path between *s* and *d*.

| а    | N( <b>a</b> ) | <i>h</i> for $\gamma(\boldsymbol{a}, \boldsymbol{n}) \cdot P_h(\boldsymbol{n})$ where $n \in N(\boldsymbol{a})$ |      |      |      |      |  |
|------|---------------|---|------|------|------|------|--|
|      |               | 0   | 1    | 2    | 3    | 4    |  |
| 0000 | 0001          | 1.00  | 1.00 | 0.96 | 1.00 | 1.00 |  |
|      | 0010          | 1.00  | 1.00 | 0.96 | 0.99 | 1.00 |  |
|      | 0100          | 1.00  | 0.75 | 0.96 | 0.99 | 1.00 |  |
|      | 1000          | 0.00  | 0.00 | 0.00 | 0.00 | 0.00 |  |
| 0001 | 0000          | 0000  | 1.00 | 0.75 | 0.96 | 0.96 |  |
|      | 0011          | 0011  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0101          | 0101  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 1001          | 1001  | 1.00 | 0.75 | 0.96 | 0.99 |  |
| 0010 | 0011          | 0011  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0000          | 0000  | 1.00 | 0.75 | 0.96 | 0.96 |  |
|      | 0110          | 0110  | 1.00 | 0.50 | 0.83 | 0.99 |  |
|      | 1010          | 1010  | 1.00 | 1.00 | 0.88 | 1.00 |  |
| 0011 | 0010          | 0010  | 1.00 | 1.00 | 0.96 | 0.99 |  |
|      | 0001          | 0001  | 1.00 | 1.00 | 0.96 | 1.00 |  |
|      | 0111          | 0111  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 1011          | 1011  | 1.00 | 0.75 | 1.00 | 0.99 |  |
| 0100 | 0101          | 0101  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0110          | 0110  | 0.00 | 0.00 | 0.00 | 0.00 |  |
|      | 0000          | 0000  | 1.00 | 0.75 | 0.96 | 0.96 |  |
|      | 1100          | 1100  | 1.00 | 1.00 | 0.88 | 1.00 |  |
| 0101 | 0100          | 0100  | 1.00 | 0.75 | 0.96 | 0.99 |  |
|      | 0111          | 0111  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0001          | 0001  | 1.00 | 1.00 | 0.96 | 1.00 |  |
|      | 1101          | 1101  | 1.00 | 1.00 | 1.00 | 0.99 |  |
| 0110 | 0111          | 0111  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0100          | 0100  | 0.00 | 0.00 | 0.00 | 0.00 |  |
|      | 0010          | 0010  | 1.00 | 1.00 | 0.96 | 0.99 |  |
|      | 1110          | 1110  | 0.00 | 0.00 | 0.00 | 0.00 |  |
| 0111 | 0110          | 0110  | 1.00 | 0.50 | 0.83 | 0.99 |  |
|      | 0101          | 0101  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 0011          | 0011  | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 1111          | 1111  | 1.00 | 1.00 | 0.96 | 1.00 |  |

TABLE I: AN EXAMPLE OF ROUTING PROBABILITIES IN  $Q_4$ 



Fig. 4. A 4-dimensional Hypercube  $Q_4$  with faulty links.

## V. FAULT-TOLERANT ROUTING ALGORITHM

In this section, we show an algorithm to find a fault-free path in a hypercube with a set of faulty links. Routing is based on the routing probabilities of neighbor nodes stored in each node.

In an n-dimensional hypercube with a set of faulty links, we assume that each node a stores the routing probabilities of all neighbor nodes  $n \ (\in N(a))$  with respect to all Hamming distances  $h \ (0 \le h \le n)$ . Then, for a source node s and a destination node d, we propose a stochastic routing algorithm (SRA) that establishes a fault-free path between them. Algorithm SRA takes a current node c and a destination node d as its arguments. In the initial state, SRA is called with a source node and a destination node. The final result will be a

| а    | N(a) | $\tilde{h}$ for $\gamma(\boldsymbol{a}, \boldsymbol{n}) \cdot P_h(\boldsymbol{n})$ where $n \in N(\boldsymbol{a})$ |      |      |      |      |  |
|------|------|--|------|------|------|------|--|
|      |      | 0  | 1    | 2    | 3    | 4    |  |
| 1000 | 1001 | 1.00   | 0.75 | 0.96 | 0.99 | 1.00 |  |
|      | 1010 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 1100 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 0000 | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 |  |
| 1001 | 1000 | 1.00   | 0.75 | 0.96 | 0.94 | 1.00 |  |
|      | 1011 | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 |  |
|      | 1101 | 1.00   | 1.00 | 1.00 | 0.99 | 1.00 |  |
|      | 0001 | 1.00   | 1.00 | 0.96 | 1.00 | 1.00 |  |
| 1010 | 1011 | 1.00   | 0.75 | 1.00 | 0.99 | 1.00 |  |
|      | 1000 | 1.00   | 0.75 | 0.96 | 0.94 | 1.00 |  |
|      | 1110 | 1.00   | 0.75 | 1.00 | 0.94 | 1.00 |  |
|      | 0010 | 1.00   | 1.00 | 0.96 | 0.99 | 1.00 |  |
| 1011 | 1010 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 1001 | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 |  |
|      | 1111 | 1.00   | 1.00 | 0.96 | 1.00 | 1.00 |  |
|      | 0011 | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 |  |
| 1100 | 1101 | 1.00   | 1.00 | 1.00 | 0.99 | 1.00 |  |
|      | 1110 | 1.00   | 0.75 | 1.00 | 0.94 | 1.00 |  |
|      | 1000 | 1.00   | 0.75 | 0.96 | 0.94 | 1.00 |  |
|      | 0100 | 1.00   | 0.75 | 0.96 | 0.99 | 1.00 |  |
| 1101 | 1100 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 1111 | 1.00   | 1.00 | 0.96 | 1.00 | 1.00 |  |
|      | 1001 | 1.00   | 0.75 | 0.96 | 0.99 | 1.00 |  |
|      | 0101 | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 |  |
|      | 1111 | 1.00   | 1.00 | 0.96 | 1.00 | 1.00 |  |
| 1110 | 1100 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 1010 | 1.00   | 1.00 | 0.88 | 1.00 | 0.99 |  |
|      | 0110 | 0.00   | 0.00 | 0.00 | 0.00 | 0.00 |  |
| 1111 | 1110 | 1.00   | 0.75 | 1.00 | 0.94 | 1.00 |  |
|      | 1101 | 1.00   | 1.00 | 1.00 | 0.99 | 1.00 |  |
|      | 1011 | 1.00   | 0.75 | 1.00 | 0.99 | 1.00 |  |
|      | 0111 | 1.00   | 1.00 | 1.00 | 1.00 | 1.00 |  |

faulty-free path between node *s* and node *d*.

Algorithm SRA, first, calculates the Hamming distance h=H(c, d). If h = 0, the message is delivered to the current node c, and the algorithm terminates. Otherwise, that is, if h>0, algorithm SRA looks up the set of preferred neighbor nodes of the current node for the destination node to find the node  $n_0^*$  that has the maximum value of routing probabilities with respect to (H(c, d) - 1). If the routing probability is positive, the message is sent to the node.

procedure SRA(c, d) begin h := H(c, d); if h = 0 then begin deliver the message to c; exit end;  $n_0^* := \arg \max_{n \in N_0(c, d)} \{ (c, n)P_h \ 1(n) \}$ ; if  $\gamma(c, n_0^*)P_h \ 1(n_0^*) > 0$  then begin SRA( $n_0^*, d$ ); exit end;  $n_1^* := \arg \max_{n \in N_1(c, d)} \{ (c, n) P_h \ 1(n) \}$ ; SRA( $n_1^*, d$ ) end

Fig. 5. Stochastic routing algorithm SRA.

If all of the routing probabilities of preferred neighbor nodes are zero, the spare neighbor nodes are checked to find the node  $n_1^*$  that has the maximum routing probability with respect to (H(c, d) + 1), and the message is sent to the node.

Fig. 5 shows a pseudo code for the algorithm where the exception handling for the case of h = n is omitted. From theorem of existence, if a fault-free path between the current node and the destination node exists,  $P_{h-1}(\mathbf{n}_0^*) > 0$  or  $P_{h+1}(\mathbf{n}_1^*) > 0$  holds. Therefore, routing always fails with an infinite loop.

#### VI. PERFORMANCE EVALUATION

#### A. Time and Space Complexity

As mentioned above, each node a runs a program based on Procedure RP shown in Fig. 5 simultaneously, and calculates routing probabilities in synchronization with neighbor nodes. In this sub section, we estimate the time and space complexities consumed at each node. We assume that it takes O(1) time to send a value of routing probability to one neighbor node. In addition, we assume that a value of routing probability can be stored in one machine word.

The time complexity of Procedure RP at a node a in  $Q_n$  is  $O(n^2 \log n)$ .

(**Proof**) It takes O(1) time to calculate  $P_0(a)$  in Procedure RP at a node a, and it takes O(n) time to send it to all neighbor nodes. To obtain  $P_h(a)$  for a Hamming distance h  $(1 \le h \le n)$ , it is necessary to receive  $\gamma(a, n) P_{h-1}(n)$ , sort them in ascending order, and calculate  $\sum_{k=1}^{n} \frac{1}{k-l}C_{h-l}/nC_h$ . These operations take O(n),  $O(n \log n)$ , and O(n) time.

Therefore, to obtain  $P_h(a)$  for all h  $(1 \le h \le n)$ , it takes  $O(n^2 \log n)$  time. Finally, it takes O(n) time to receive  $\gamma(a, n)$   $P_n(n)$  from all neighbor nodes n. In total, the time complexity of Procedure RP is  $O(n^2 \log n)$ .

In addition, the space complexity of Procedure RP at a node a in  $Q_n$  is  $O(n^2)$ .

(**Proof**) At each node a, it is necessary to store  $P_h(n)$  for all neighbor nodes  $n(\in N(a))$  and all Hamming distances h  $(0 \le h \le n)$ . Therefore, the space complexity of Procedure RP is  $O(n^2)$  in total.

#### B. Computer Experiment

```
procedure GRA (c, d)
begin
  h := H(c, d);
  if h = 0 then begin
    deliver the message to c;
    exit
  end;
  if \exists n_0^* \in N_0(c, d) and \gamma(c, n_0^*) = 1 then begin
    GRA (n_0, d);
    exit
  end;
  if \exists n_1^* \in N_1(c, d) and \gamma(c, n_1^*) = 1 then begin
    GRA (n_1^*, d);
    exit
  end;
  error("Routing Failed")
end
```

Fig. 6. Greedy routing algorithm GRA.

In this section, we give the detail of the result of a computer experiment we conducted to evaluate our algorithm. We carried out the computer experiment in  $Q_n$  (n = 7, 8, 9, 10) with the ratios of faulty links  $\alpha$  from 0.0 to 0.9. We measured the ratios of successful routings and the path lengths of our stochastic routing algorithm SRA with a simple greedy algorithm GRA shown in Fig. 6.

Specifically, we first randomly selected faulty links in  $Q_n$  with the ratio of  $\alpha$ . Next, we selected a source node s and a destination node d randomly. Finally, after checking the connectivity of s and d, we applied the link-fault-tolerant routing algorithm. If s and d are not connected, that is, if there is no fault-free path between them, we restarted from the selection of faulty links. For each pair of n and  $\alpha$ , we conducted at least 100,000 times of attempts.

Fig. 7-Fig. 10 show the ratios of successful routings in  $Q_7$ ,  $Q_8$ ,  $Q_9$ , and  $Q_{10}$ , respectively, by our link-fault- tolerant routing algorithm SRA and the greedy routing algorithm GRA. In addition, Fig. 11-Fig. 14 show the average path lengths in  $Q_7$ ,  $Q_8$ ,  $Q_9$ , and  $Q_{10}$ , respectively. From Fig. 7-Fig. 10, we can observe that in  $Q_7$ ,  $Q_8$ ,  $Q_9$ , and  $Q_{10}$ , the ratios of successful routings of SRA are improved by at most 0.20 ( $\alpha = 0.39$ ), 0.20 ( $\alpha = 0.43$ ), 0.21 ( $\alpha = 0.43$ ), and 0.21 ( $\alpha = 0.41$ ), respectively, compared to GRA.







Fig. 8. Ratio of successful routings in  $Q_8$  with faulty ratio of Links  $\alpha$ .

Note that, in all the figures, if the ratios of faulty links exceed certain criteria around 0.8, trivial pairs of the source and destination nodes are frequently adopted and the ratios of successful routings increase rapidly. Also, from Fig. 11-Fig. 14, we can observe that in  $Q_7$ ,  $Q_8$ ,  $Q_9$ , and  $Q_{10}$ , the average path lengths of SRA are increased by at most 0.786 ( $\alpha = 0.70$ ),

0.946 ( $\alpha = 0.76$ ), 1.14 ( $\alpha = 0.77$ ), and 1.32 ( $\alpha = 0.80$ ), respectively, compared to GRA.



Fig. 9. Ratio of successful routings in  $Q_9$  with faulty ratio of links  $\alpha$ .



Fig. 10. Ratio of successful routings in  $Q_{10}$  with faulty ratio of links  $\alpha$ .



Fig. 11. Average path lengths for successful routing in  $Q_7$  with faulty ratio of links  $\alpha$ .



Fig. 12. Average path lengths for successful routing in  $Q_8$  with faulty ratio of links  $\alpha$ .

As mentioned above, if the ratios of faulty links exceed certain criteria around 0.8, trivial pairs of the source and destination nodes are frequently adopted and the average path lengths also decrease rapidly.

Consequently, we can conclude that our stochastic linkfault-tolerant routing algorithm SRA attained high ratios of successful routings with a small cost.



Fig. 13. Average path lengths for successful routing in  $Q_9$  with faulty ratio of links  $\alpha$ .



Fig. 14. Average path lengths for successful routing in  $Q_{10}$  with faulty ratio of links  $\alpha$ .

# VII. CONCLUSION

We have proposed a stochastic link-fault-tolerant routing algorithm that finds a fault-free path in a hypercube with a set of faulty links. The algorithm collects information called routing probabilities from neighbor nodes to make use of them for routing. We have conducted a computer experiment in which we compared our algorithm with a simple greedy algorithm. As a result, we have verified the effectiveness of our algorithm.

Now, we are interested in applying our approach to variants of a hypercube or hierarchical interconnection networks based on a hypercube.

#### REFERENCES

- C. L. Seitz, "The cosmic cube," *Communications of the ACM*, vol. 28, no. 1, pp. 22-33, Jan. 1985.
- [2] J. L. Gustafson, S. Hawkinson, and K. Scott, "The architecture of a homogeneous vector supercomputer," *Journal of Parallel and Distributed Computing*, vol. 3, no. 3, pp. 297-304, Sept. 1986.
- [3] J. P. Hayes, T. Mudge, Q. F. Stout, S. Colley, and J. Palmer, "A microprocessor-based hypercube supercomputer," *IEEE Micro*, vol. 6, no. 5, pp. 6-17, Oct. 1986.

- [4] J. C. Peterson, J. O. Tuazon, D. Lieberman, and M. Pniel, "The mark iii hypercube-ensemble concurrent computer," in *Proc. the 1985 International Conference on Parallel Processing*, 1985, pp. 71-73.
- [5] J. Rattner, "Concurrent processing: A new direction in scientific computing," in *Proc. the National Computer Conference*, July 1985, pp. 157-166.
- [6] J. O. Tuazon, J. C. Peterson, M. Pniel, and D. Lieberman, "Caltech /JPL mark ii hypercube concurrent processor," in *Proc. the 1985 International Conference on Parallel Processing*, 1985, pp. 666-673.
- [7] Top500.org, Top500 list June 2014. [Online]. Available: http://top500.org/list/2014/06/
- [8] K. Ghose and K. R. Desai, "Hierarchical cubic networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, no. 4, pp. 427-435, Apr. 1995.
- [9] Y. Li and S. Peng, "Dual-cube: A new interconnection network for high-performance computer clusters," in *Proc. the International Computer Symposium*, Dec. 2000, pp. 51-57.
- [10] Y. Li, S. Peng, and W. Chu, "A new interconnection network for large scale parallel systems," *Australian Computer Science Communications*, vol. 24, no. 3, pp. 29-36, Jan. 2002.
- [11] Q. M. Malluhi and M. A. Bayoumi, "The hierarchical hypercube: A new interconnection topology for massively parallel systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 1, pp. 17-30, Jan. 1994.
- [12] F. P. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Communications of ACM*, vol. 24, no. 5, pp. 300-309, May 1981.
- [13] Z. Jiang and J. Wu, "A limited-global information model for fault-tolerant routing in dual-cube," *The International Journal of Parallel, Emergent and Distributed Systems*, vol. 21, no. 1, pp. 61-77, Feb. 2006.
- [14] G.-M. Chiu and S.-P. Wu, "A fault-tolerant routing strategy in hypercube multicomputers," *IEEE Transactions on Computers*, vol. 45, no. 2, pp. 143-155, Feb. 1996.
- [15] G.-M. Chiu and K.-S. Chen, "Use of routing capability for fault-tolerant routing in hypercube multicomputers," *IEEE Transactions on Computers*, vol. 46, no. 8, pp. 953-958, Aug. 1997.
- [16] J. Wu, "Adaptive fault-tolerant routing in cube-based multicomputers using safety vectors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 4, pp. 322-334, Apr. 1998.
- [17] K. Kaneko and H. Ito, "Fault-tolerant routing algorithms for hypercube interconnection networks," *IEICE Transactions on Information and Systems*, vol. E84-D, no. 1, pp. 121-128, Jan. 2001.
- [18] J. Al-Sadi, K. Day, and M. Ould-Khaoua, "Fault-tolerant routing in hypercubes using probability vectors," *Parallel Computing*, vol. 27, no. 10, pp. 1381-1399, Sept. 2001.
- [19] J. Al-Sadi, K. Day, and M. Ould-Khaoua, "Probability-based fault-tolerant routing in hypercubes," *The Computer Journal*, vol. 44, no. 5, pp. 368-373, 2001.

- [20] D. T. Duong and K. Kaneko, "Fault-tolerant routing algorithms based on approximate routable probabilities for hypercube networks," in *Proc. the 2011 International Conference on Parallel and Distributed Processing Techniques and Applications*, vol. 1, July 2011, pp. 116-122.
- [21] D. T. Duong and K. Kaneko, "Fault-tolerant routing based on approximate directed routable probabilities for hypercubes," *Future Generation Computer Systems*, vol. 37, pp. 88-96, July 2014.



Lam Boi Ngoc was born on August 5, 1993 in Ho Chi Minh City, Viet Nam. She has been studying in Ho Chi Minh City University of Science, Viet Nam since September 2011. From 2013 to 2014, she was studying in Kaneko Laboratory, Tokyo University of Agriculture and Technology, Tokyo, Japan. Currently, she is working on graph theory, routing probability, and fault tolerance.



**Bui Thi Thuan** was born on February 16, 1990 in Hung Yen, Viet Nam. She received the B.E. degree from Hanoi University of Science and Technology, Viet Nam in 2013. Currently, she is a master student of Graduate School of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan. Her main research areas are graph and network theory, interconnection networks, and dependable computing.



Yuki Hirai was born on November 11, 1983 in Kanagawa, Japan. He received the B.L.A. and M.Ed. degrees from Tokyo Gakugei University in 2007 and 2009, respectively. He also received the Ph.D. degree from University of Tsukuba in 2012. Now, he is an Assistant Professor at the Institute of Engineering in Tokyo University of Agriculture and Technology. His current research areas are educational technologies and human-computer interaction. He is a member of

IPSJ and JSAI.



Keiichi Kaneko was born on November 13, 1962 in Tokyo, Japan. He received the B.E., M.E. and Ph.D. degrees from the University of Tokyo in 1985, 1987 and 1994, respectively. Now, he is a Professor at the Institute of Engineering and the Manager of Information Media Center at Tokyo University of Agriculture and Technology. His main research areas are functional programming, parallel and distributed computation, partial evaluation, and dependable

systems. He is a member of IEEE, ACM, IEICE, IPSJ and JSSST.